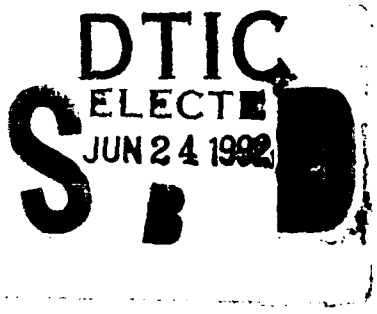
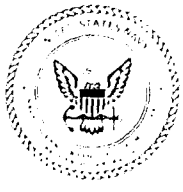
Survey of Texture Segmentation, Classification, and Synthesis Methods



B. S. Bourgeois
C. L. Walker
Mapping, Charting and Geodesy Division
Ocean Science Directorate



Foreword

This report summarizes the significant literature in the area of image texture segmentation, classification, and synthesis. The intent is to provide guidance and direction to the approaches available for image texture processing and a measure of their relative merit. The goal of this effort is to utilize texture processing techniques for the classification of acoustic provinces in sidescan sonar imagery.

William B Moseley

W. B. Moseley
Technical Director

L. R. Elliott

L. R. Elliott, Commander, USN
Commanding Officer

Executive Summary

This report reviews the literature in the areas of image texture segmentation, classification, and synthesis methods. The approaches to these areas are grouped into areas of fractal, spline, neural networks, modeling, and stochastic methods. An immense amount of literature in the area was reviewed, and techniques with the most merit are presented. From the review it appears that no single approach provides a robust texture analysis methodology without requiring overwhelming complexity.

Acknowledgments

The authors thank the Office of Naval Technology for funding this project under Program Element number 0602435N, managed by CDR Lee Bounds, ONT, Code 228T. We also thank Dr. Herbert Eppert, Jr., Director, Ocean Science Directorate, and Mr. Michael Harris, Head of the Mapping, Charting, and Geodesy Division, for their interest and support of this project.

The mention of commercial products or the use of company names does not in any way imply endorsement by the U.S. Navy or NOARL.

Contents

1.0	**Introduction**	1
2.0	**Technique Comparisons**	3
3.0	**Fractal Techniques**	5
4.0	**Spline Techniques**	10
5.0	**Neural Networks**	11
6.0	**Modeling and Stochastic Techniques**	14
7.0	**Conclusions**	17
8.0	**References**	18

Survey of Texture Segmentation, Classification, and Synthesis Methods

1.0 Introduction

This report surveys the documents reviewed in the areas of texture segmentation, classification, and synthesis. The motivation behind this research is to obtain an inventory of the techniques available, and their relative merit. The purpose of this research is to assemble a core of techniques that can be applied to generic texture processing problems. The goal of this effort is to utilize texture processing techniques for the classification of acoustic provinces in sidescan sonar imagery. This section defines texture and discusses some of the perceptions obtained from the literature review.

Modestino et al.[1] describe tone as the average gray level of a region and texture as the spatial distribution of gray levels in a region. They further indicate that a subtle relationship exists between tone and texture that is highly dependent upon resolution; tone dominates at both high and low resolution of a scene. They define texture as "a basic local order or quasi-homogeneous pattern that is repeated in a nearly periodic manner over some region large in comparison to the local pattern size." They indicate, and it seems to be widely accepted, that there are two fundamental approaches to texture discrimination: structural and statistical.

In general, a given texture problem requires a combination of these techniques. Consider the following examples of textures. A sand texture can be handled with strictly statistical methods and a group of parallel lines with strictly structural methods. Brick has a kernel that can best be described by statistical methods, but the placement of this kernel is best handled by structural methods. Straw has a structurally defined kernel with a statistical placement rule.

In reviewing the literature about image texture, a perception develops that there is pursuit for an algorithm that will globally perform the task of texture analysis. In the two to three decades of this research, hundreds of researchers have brought their own special tools into the field from their areas of expertise. As a consequence, this area is difficult to enter; there is an abundance of diverse techniques, each with its own special language and mathematics. Furthermore, each approach in turn appears to suffer a similar fate. In the beginning each new technique is typically proclaimed as a definitive solution, but shortcomings are quickly discovered. An abundance of research then ensues to overcome these shortcomings. However, in the process of this fortification, the technique often becomes too complicated and unwieldy to use. The most recent example of this process is demonstrated by the relatively new area of fractals. This approach, attributed to Mandelbrot,[2] is currently in the process of fortification, as evidenced by the efforts of Ait-Kheddache and Rajala.[3] In this 1988 paper they indicate that visually distinct textures may be indiscernible by fractal dimension (e.g., bark and pigskin) and proposes the use of "higher order" fractals for the segmentation

and classification of texture, based on work done by Hentschel and Procaccia.[4]

It would seem that the efforts in the area of texture could be best described as a group of techniques N_k, which are completely effective on a corresponding group of images I_k, where each I_k is a subset of the whole image space Ω. Typically, each N_k technique is straightforward and computationally efficient, and is effective on approximately 80% to 90% of the images to which it is applied. In an attempt to modify N_k so that it can also contend with images in the set $\Omega - I_k$, the technique quickly grows to be computationally inefficient, complicated, and unrecognizable. An explanation for this recurring phenomenon may lie in the fact that the human vision/recognition system is not well understood, and the efforts in the way of texture processing are an attempt to model and mimic this system.

The first stages of this literature review have established that the best method of texture analysis is, in fact, several methods. Almost any particular texture analysis technique can likely be made to handle most images, but the result is an overly complicated, time consuming method. It would therefore seem more practical to use a variety of methods in their simplest and most computationally efficient form as a front end to the human vision system or perhaps to an artificial intelligence (AI) system. Pursuing avenues that lead to complex and inefficient solutions is needless, since many already exist and the computer hardware necessary to handle these solutions does not. Review of the literature has proven fruitful in this regard: it reveals blind avenues that others have followed and provides a menu of techniques with various attributes. Each technique should be judged primarily on its efficiency and simplicity, and different techniques should be compared in regard to the particular class of imagery on which they are effective.

A new technique for texture processing has recently emerged. Neural networks have shown a great propensity for easily coping with nonlinear and chaotic phenomena. Neural networks are conceptually and computationally straightforward, but presently require excessive processing time when simulated on a conventional digital computer. There is a great amount of interest in the field, and several vendors report that specialized hardware will soon be available. Widrow and Winter[5] have already used neural networks to create a pattern recognition classifier that is insensitive to translation, rotation, and changes in scale; these operations have posed significant problems in the area of image processing, yet are easily coped with by the human visual system. Neural networks may prove to be highly effective for making decisions based on information from several different texture extraction preprocessing algorithms, or it may also be possible to directly apply a neural network to model the phenomena that generate an image.

The following sections review the more promising techniques and indicate their relative merit. Section 2.0 reviews papers that have done comparisons between various algorithms. Section 3.0 reviews fractal approaches, and Section 4.0 discusses a paper on the generation of fractals using spline techniques. Section 5.0 discusses the newer neural network approaches to texture processing, and Section 6.0 briefly reviews some of the more traditional modeling and stochastic techniques. Finally,

section 7.0 discusses the principal conclusions drawn from this literature review.

2.0 Technique Comparisons

Surprisingly few papers compare various techniques or attempt to survey the area of texture processing. Furthermore, most textbooks provide only a cursory coverage of the wide variety of the techniques that have been attempted.

Perhaps this deficiency is best explained, as it has been so aptly put by many, by the fact that so many of these techniques are ad hoc by nature. Only two survey papers were found: Haralick,[6] and Conners and Harlow.[7] Haralick is referenced many times, and his 1979 paper seems to be well accepted as a baseline summary of the available techniques up to that time. Haralick made a significant point in referring to the current techniques, in that they typically emphasized either the tonal primitive properties or their spatial interrelations, but not both. In this paper he indicated eight popular statistical approaches to texture analysis:

autocorrelation function — Given a bounded region, $0 \le u \le L_x$ and $0 \le v \le L_y$, where (x, y) are the x, y translations, and $I(u, v)$ is the total "energy" of the image at position (u, v), then the autocorrelation is given by:

$$\rho(x,y)=\frac{1}{(L_x-|x|)(L_y-|y|)}\;\frac{\int\int_{-\infty}^{\infty} I(u,v)I(u+x,v+y)\,du\,dv}{\frac{1}{L_xL_y}\int\int_{-\infty}^{\infty} I^2(u,v)\,du\,dv}. \qquad (1)$$

Haralick describes tonal primitives as regions with uniform tonal properties. If the tonal primitives are large, the autocorrelation drops off slowly with distance; if small it drops off rapidly. If the tonal primitives are periodic, the autocorrelation will also display a periodic behavior.

optical transforms — the light amplitude distribution at the front and rear focal planes of a lens are Fourier transforms of one another.

digital transforms — images are typically divided into smaller areas, and digital transforms are applied. The areas are then compared based on the transform characteristics. Popular transforms include the discrete Fourier transform (DFT), sine and cosine transforms, Hadamard transform, Slant transform and the Harr transform.

textural edginess — fine textures have many edges per unit area.

structural elements — for binary images, this technique emphasizes the shape aspects of tonal primitives.

spatial gray-tone co-occurrence probabilities — a coarse texture has a slight distribution change with distance, and a fine texture has a larger change. This method does not capture shape aspects of the tonal primitives and does not work well for textures with large area primitives.

A previous paper by Haralick et al.[8] showed how to obtain 14 different features from the gray-tone co-occurrence matrix, including entropy, maximum probability, contrast, correlation, inverse difference moment, and probability of run length. The co-occurrence matrix P, $\{P : G \times G$ to $[0,1]\}$ for an image I and binary relation R is given by:

$$P(i,j) = \frac{\text{no.}\left[((a,b),(c,d)) \in R \mid I(a,b) = i \text{ and } I(c,d) = j\right]}{\text{no. } R} \qquad (2)$$

gray-tone run lengths — primitives for this method are maximal collinear connected sets of the same gray tone. A coarse texture will have many pixels in a gray-tone run, and a fine texture will have fewer.

autoregressive models — these models utilize linear estimates of previous gray tones to generate the next gray tone. These models may be causal, semicausal, or anticausal, and often inject a noise variable. For coarse textures the coefficients will be similar; for fine textures the coefficients in the linear estimate will have a wide variation. This approach is simple and easy, but does work well with macro textures.

Haralick[6] further indicated that the first three techniques measure spatial frequency, but that they are not invariant under monotonic transformations of gray tone. For these techniques, fine textures will have high frequencies, and coarse textures will have predominantly low frequencies. He further mentioned that Weszka et al.[9] showed that the effectiveness of these techniques were significantly poorer than other approaches.

Haralick made only little mention of structural approaches in this paper. One such technique is mosaics, where a picture is tessellated into regions and gray levels are assigned to each region based on a specified probability density function. Also, he indicated that for macro textures, investigators are using histograms of primitive properties and co-occurrence of primitive properties as a generalization of the structural and statistical approaches.

Conners and Harlow[7] made a detailed comparison of four algorithms:

- spatial gray-level dependence method (SGLDM)
- gray-level run length method (GLRLM)
- gray-level difference method (GLDM)
- power spectrum (PSM).

Conners and Harlow examined these algorithms based on the texture information content of the intermediate matrices of the processes, and thus determined the relative sets over which each algorithm is effective. The results of their analysis follow:

- GLRLM could not discriminate all visually distinct texture pairs.
- There exists a visually distinct texture pair for which GLDM cannot discriminate for any value of spacing, d.
- PSM cannot discern all visually distinct texture pairs.
- SGLDM can discern a larger class of textures than GLRLM, even when only a sample spacing of 1 is used. SGLDM is also more powerful than GLDM and PSM.
- GLDM is more powerful than PSM.

• None of these algorithms could discriminate between a Markov texture and a 180 degree rotation of the texture.

• The GLRLM suffers from noise sensitivity.

• SGLDM and GLDM work better when multiple intersample spacing distances are used.

• SGLDM is much more powerful than the PSM.

Weszka et al.[9] analyzed these same algorithms, but Conners and Harlow[7] indicate that the comparison methods used by Weszka et al. were not general enough to completely judge the algorithm's performance. However, both papers drew the same conclusions on the relative power of these four algorithms.

3.0 Fractal Techniques

A fractal is an object with the property that it is self-similar at various scales of magnification. Consequently, such an object has a fractional dimension and its power spectrum is a function of 1/frequency. The classical example[10] is the measurement of a shoreline, where the length of the shoreline increases as the length of the measuring device decreases. This effect can be stated as follows: given a yardstick of length L, the measurement of an n-dimensional surface is given by $M = nL^D$, where D is the topological dimension of the yardstick. Given a fractal surface, D is the fractional power that yields a consistent measure M for all sizes of the yardstick L.[10] A more rigorous definition is: A random function $I(x)$ is a fractal Brownian function if for all x and Δx

$$\text{Prob.}\left(\frac{I(x+\Delta x)-I(x)}{\|\Delta x\|^H} < y\right) = F(y), \tag{3}$$

where the spectral density of a fractal Brownian function is proportional to f^{-2H-1}.

Fractals and fractal dimension have proven to be extremely valuable in texture analysis and synthesis.[10,11,12,13] They have been the only effective method for generating realistic-looking terrain, clouds, and many other objects that occur in nature. The use of fractal dimension has proven to be highly effective in the segmentation and classification of many textures. Perhaps their greatest attribute is the simplicity and numerical efficiency of their algorithms. Mandelbrot[2] is acknowledged to be the father of fractal mathematics, although his work may be hard to follow and utilize. Books by Barnsley et al.,[11] Barnsley,[12] and Peitgen and Richter[13] seem to be much more functional. *The Science of Fractal Images*[11] is a collection of papers and lecture notes from many of the major researchers in the area, and includes several ready-to-implement algorithms. *Fractals Everywhere*[12] is an excellent treatment of the mathematics involved and is easy to follow. *The Beauty of Fractals*[13] takes more of a dynamical systems approach, looking deeper into the chaotic phenomena that generate fractal shapes.

In his paper, Brammer[14] discusses many of the uses and characteristics of fractals and fractal dimension.

• It is generally true that the image of a fractal set is fractal with a direct relationship between the dimensions.

• The fractal dimension at multiple resolutions can be used to determine object ranges in imagery.

• The fractal sum of pulses method has been used for cloud analysis and forecasting.

• Fractal dimension is used for detecting manmade objects in natural scenes and for the automated detection of cracks in industrial applications.

• Fractal techniques are being used for image compression. The Peano scan method achieves a bit rate of less than one bit per pixel. Barnsley et al.[11] are working on an iterated function method of image compression that would give supervised compression ratios on the order of 10,000:1, and automated ratios of 125:1.

Fournier et al.[15] provide algorithms that are a modification of Mandelbrot's techniques, but are more computationally efficient. They explain that the shear displacement process requires $O(N^3)$ operations, that the modified Markov process requires $O(N \log(N))$ operations, and that the inverse Fourier transform requires $O(N \log(N))$ operations. Fournier et al. provide algorithms in Pascal for fractal line and surface generation that require only $O(N)$ operations. They note that the generated objects are not stationary, isotropic, or self-similar, but that they are realistic looking.

Pentland[10] is frequently referenced by other researchers, and in this paper he derives the relationship between the fractal dimension of a surface and the fractal dimension of its image. He shows that a three-dimensional (3-D) surface with a spatially isotropic fractal Brownian shape produces an image whose intensity surface is fractal Brownian and whose fractal dimension is identical to that of the components of the surface normal, *given a Lambertian surface reflectance and constant illumination and albedo.* Thus, if the surface is fractal, so will be the image. Furthermore, the fractal dimension of imaged contours is the same as that of the 3-D contour, and the surface's dimension is 1 plus the contour's dimension. To determine if an image exhibits a fractal nature, Pentland suggests that histograms over multiple scales can be used. Using this method, if the standard deviation of the histograms are nearly linear versus scale, then the image is fractal-like. An important characteristic noted in this paper is that the fractal dimension of regions that contain a boundary is typically less than the topological dimension. Although this result is erroneous,[10] it can be used successfully for edge detection in an image.

In this paper, Pentland also reports the results of a fractal dimension texture segmenter, which yielded a classification accuracy of 84.4% in contrast to 65% for correlation statistics and 72% for co-occurrence statistics. He used the power spectrum method of determining fractal dimension given by $\log(P(f)) = -(2H + 1)\log(f) + k$, where $P(f)$is the power spectrum, and the fractal dimension is given by $2 - H$. In this test he used 8 • 8 pixel blocks, and he indicated that the results typically proved effective over scales of 4:1 and in some cases as much as 8:1. Pentland also indicated that one of the shortfalls of fractals is that they do not describe regular or large-scale spatial structures. However, a possible method for handling this situation is first to detect the edges

in the image and then to analyze the nonedge regions. He also points out that fractals are strictly an abstraction and that physical objects will behave like fractals only over a range of parameters; researchers seem to have often overlooked this point in their discussions of the use of fractal dimension for image analysis.

Keller et al.[16] describe an unsupervised, low computational segmentation routine that uses an improvement on the lacuarity feature introduced by Mandelbrot. The term lacuarity is used to describe the characteristic of fractals that have the same fractal dimension but exhibit different textures. Keller et al. indicate that fractal dimension alone is generally insufficient to classify natural textures, and that natural fractal surfaces typically exhibit statistical self-similarity vice deterministic self-similarity. In this paper, the segmentation was performed by clustering, and the box dimension was used to estimate the fractal dimension. The paper includes the algorithms for calculating the box dimension and the technique for K-means segmentation. Keller modified the box-counting method for dimension estimation, thereby producing the "interpolation" method. This method helped to overcome the quantization effects by interpolating between the center point of a cube and each of its neighbors. The results of the segmentation were excellent, indicating that this technique shows great promise.

Margerum and Werkheiser[17] provide code written in LISP for generating fractal landscapes. They included several generated images and demonstrated the effects of parameter change in the generation algorithm. The scene generation involved using a coarse elevation grid and the formula $E_{new} = E_{avg} + Rd^{3-D}$, where E_{new} is the new elevation of the center point of a square on the grid, E_{avg} is the average of the four corner values of the square, R is a random number from a Gaussian distribution, d is the distance from the center point of the square to a vertex, and D is the fractal dimension. A fractal dimension near 2 provided the most realistic-looking results.

Vernazza[18] provided a much needed comparison of fractal-dimension estimators. He compared the results of four approaches to measuring the fractal dimension of an image: the spatial Pentland approach, the frequential Pentland approach (using the power spectrum), the blanket approach, and the Euclidean approach. For the test he used an image with fractal dimension of 1.2 and an $8 \cdot 8$ mask size. Once the fractal dimension of each block was computed, the image was segmented using a histogram of the fractal dimension. The best results were given by the spatial Pentland method and the blanket approach, with the blanket approach providing the best estimate of fractal dimension. The frequential Pentland approach and the Euclidean approach produced very jagged curves, requiring linear estimation to determine the fractal dimension. The procedure for the blanket approach is as follows:

An image is covered with an upper surface $u(\varepsilon, i, j)$ and a lower surface $b(\varepsilon, i, j)$. Defining the gray-level image $g(i, j) = u(0, i, j) = b(0, i, j)$ at the beginning, with $\varepsilon = 1, 2, 3, \ldots$, the blanket surfaces are

$$u(\varepsilon, i, j) = \max \{u(\varepsilon - 1, i, j) + 1, \max u(\varepsilon - 1, m, n)\}$$

$$\text{for } |(m, n) - (i, j)| \leq 1 \tag{4}$$

and

$$b(\varepsilon, i, j) = \min \{b(\varepsilon - 1, i, j) - 1, \min b(\varepsilon - 1, m, n)\}$$

$$\text{for } |(m, n) - (i, j)| \leq 1 , \tag{5}$$

where the image points (m, n) are the four neighboring points of (i, j). The blanket volume is given by

$$V(\varepsilon) = \sum_{i,j} u(\varepsilon, i, j) - b(\varepsilon, i, j) \tag{6}$$

and the surface area is

$$A(\varepsilon) = (V(\varepsilon) - V(\varepsilon - 1))/2 . \tag{7}$$

Thus, the surface area A is computed for different ε values and, since $A(\varepsilon) = \lambda \varepsilon^{2-D}$, D can be computed in the bilog plane as an estimate of the linear regression.

Peleg et al.[19] introduce the concept of fractal signature. They describe the fractal signature as the change in measured image surface area with a change in scale. Using the blanket method given above by Vernazza, they define the fractal signature as $S(\varepsilon) = 2 - D$. For a truly fractal object, S is invariant with changes in ε. For a nonfractal surface, the magnitude of the fractal signature $S(\varepsilon)$ indicates the amount of information that is lost for a given yardstick ε. A high value of $S(\varepsilon)$ for a small ε indicates the presence of high frequencies in the image, and high value of $S(\varepsilon)$ for a large ε indicate the presence of low frequencies in the image.

Thus, $S(\varepsilon)$ directly provides information about the fineness or coarseness of a texture. Using the Peleg et al. method, textures are compared via their fractal signatures:

$$D(i, j) = \sum_{\varepsilon} (S_i(\varepsilon) - S_j(\varepsilon))^2 \log\left(\frac{\varepsilon + 1/2}{\varepsilon - 1/2}\right) \tag{8}$$

where the log weighting is due to the unequal spacing between points in the log-log scale. They indicate that this technique required only a small number of texture descriptors and used $\varepsilon = 2$ to 49 in this paper.

Apparently prompted by Mandelbrot's comments that coastline measurement yielded different results depending on whether the measurement is made on the land side or the water side, Peleg et al. defined a two-sided volume and area for the blanket technique. The new volume and area measures are given as

$$V_{\varepsilon}^{+} = \sum_{i,j} (u_{\varepsilon}(i, j) - g(i, j)) , \tag{9}$$

$$V_\varepsilon^- = \sum_{i,j} (g(i,j) - b_\varepsilon(i,j)), \tag{10}$$

$$A^+(\varepsilon) = V_\varepsilon^+ - V_{\varepsilon-1}^+, \tag{11}$$

$$A^-(\varepsilon) = V_\varepsilon^- - V_{\varepsilon-1}^-, \tag{12}$$

with corresponding fractal signature measures S^+ and S^-. The new distance measure for comparing textures is given by

$$D(i,j) = \sum_\varepsilon \left[\left\{ (S_i^+(\varepsilon) - S_j^+(\varepsilon))^2 + (S_i^-(\varepsilon) - S_j^-(\varepsilon))^2 \right\} \log\left(\frac{\varepsilon + 1/2}{\varepsilon - 1/2}\right) \right]. \tag{13}$$

Testing with these new measures revealed that graphs of S^- represented the shapes of objects in an image, and S^- is the same for two different images with the objects arranged in different positions. The measure S^+, which represents the background of the image, produced different results for different object arrangements. Peleg et al. also mention that these algorithms can be modified so that the blanket growth direction is directional and thus sensitive to directional textures. This paper clearly indicates that fractal measures can provide much significant information even for nonfractal images.

Arduini et al.[20] extend the Peleg et al. work by providing an adaptive method of choosing the optimal mask size for determining the fractal dimension of an image region. Arduini et al. note that a large mask will tend to smooth the fractal dimension D in the area, but a small mask will be very sensitive to noise. Furthermore, a large mask allows determination of D to a higher accuracy, but a small mask provides for better spatial resolution. They also comment that when using the blanket method, the surface area $A(\varepsilon)$ is given by Mandelbrot as $A(\varepsilon) = V(\varepsilon)/2\,\varepsilon$, and is given by Peleg et al. as $A(\varepsilon) = (V(\varepsilon) - V(\varepsilon - 1))/2$. They indicate that the Peleg et al. method is less noise sensitive and that the Mandelbrot method gives a more precise value. The Arduini et al. method involves using a $20 \cdot 20$ mask over an image, and subdividing this mask into a set of 10 smaller masks of various sizes and shapes. For each of these smaller masks, they divide the mask into $5 \cdot 5$ blocks, compute the fractal dimension for each block, and compute the variance of the fractal dimension for these $5 \cdot 5$ blocks. Thus, the best mask to use for the $20 \cdot 20$ region will be the one with lowest variance in fractal dimension among its $5 \cdot 5$ blocks.

For a test of this method, a fractal image of dimension 2.7 was used with a patch having a fractal dimension of 2.3. The patch was not visually discernible, yet the segmentation algorithm easily detected the patch. In comparison to other fractal estimation techniques using a $5 \cdot 5$ mask, a $10 \cdot 10$ mask, and a region growing technique, the Arduini et al. method provided the best results, particularly in its preservation of the edges of the patch. They indicated that the

FORTRAN-implemented algorithm took 6 hours for a 256 • 256 image on a Hewlett-Packard HP1000 computer.

Ait-Kheddache and Rajala[21] extend fractal techniques to encompass a larger class of images, i.e., to include images that contain different textures with the same fractal dimension. They use the generalized dimensions of fractals derived by Hentschel and Procaccia,[4] where the generalized dimension D_f is given by

$$D_f = \frac{1}{f-1} \lim_{l \to 0} \left[\frac{\log \left\{ \sum_{i=0} p_i \exp (f-1) \log p_i \right\}}{\log (l)} \right] \tag{14}$$

for any $f \geq 0$. Hentschel and Procaccia have proven that

$$\lim_{f \to 0} D_f = D_0 = \text{fractal dimension}, \tag{15}$$

$$\lim_{f \to 1} D_f = D_1 = \text{information dimension}, \tag{16}$$

$$\lim_{f \to 2} D_f = D_2 = \text{correlation dimension}. \tag{17}$$

Ait-Kheddache and Rajala present images of bark and pigskin as an example of textures that cannot be discerned by fractal dimension alone. In their tests they used the lower three generalized dimensions (D_0, D_1, and D_2) to classify textures. Their results were 100% classification for water/grass, water/sand, and bark/pigskin, and 81% for grass/sand. The methods for computing the information and the correlation dimension are a trifle complicated, but are explained in detail by Hentschel and Procaccia.[4]

4.0 Spline Techniques

While splines are typically a tool for graphics and for data interpolation, the paper by Szeliski and Terzopoulos[22] is pertinent to texture processing in that it provides a novel approach to generating fractal curves. Two good references for spline fundamentals are Ahlberg et al.[23] and Bartels et al.[24] Typically, a spline is a cubic function that is fit to a series of points to form a smooth curve. These functions are typically smooth in the first and second derivatives, but have a jump in the third derivative. Techniques exist for both interpolation (exact fit to a data set) and for approximation, where the maximum distance is specified between the points and the spline. The fundamental theorem of splines, due to Holladay (1957), is given by Ahlberg et al.[23]

Given a set of x_i on the interval $[a, b]$ and a corresponding set y_i, then of all functions $f(x)$ having a continuous second derivative on $[a, b]$ such that $f(x_i) = y_i$, the spline function $S(f; x)$ with junction points

at the x_i and with its second derivative equal to 0 at $x = a$, b minimizes the integral

$$\int_a^b |f''(x)|^2 \, dx \quad . \tag{18}$$

In their paper, Szeliski and Terzopoulos mention that splines are easily constrained and well suited for modeling smooth objects, and that fractals are more suitable for generating irregular shapes but are difficult to constrain. They indicate that the generation of fractals using Fourier methods results in fractals that cannot be controlled locally, and that the standard perturbation method results in a nonstationary process. To overcome these weaknesses, Szeliski and Terzopoulos employed a general class of multivariate spline models called controlled continuity splines, which afford local control over smoothness. Using this spline model, they injected white noise, and the smoothing effect of the spline spread this noise spatially to produce a fractal-like curve while retaining local control of the curve. He applied this technique to synthesize realistic terrain from sparse elevation data with very impressive results.

5.0 Neural Networks

Neural network technology is the newest player in the texture analysis arena, and it appears to have a great propensity for modeling nonlinear and chaotic phenomena. A neural network is formed by connecting one or more layers of neurons, where each neuron performs:[25]

$$X_i^{out} = g\left(\sum_j T_{ij} X_j^{inp} + \theta_i\right), \tag{19}$$

where X_j^{inp} are the inputs to the neuron, X_i^{out} is the neuron's output, T_{ij} are the neuron's weights, θ is a constant, and $g(x)$ is a nonlinear function, typically a sigmoidal form. The network is "trained" to behave like a particular system by feeding it an input/output sequence of the system to be mimicked. If t are the training output values for the p^{th} input pattern, and O are the actual output values, the network is trained by minimizing

$$E = \sum_p \sum_i \left(t_i^{(p)} - O_i^{(p)}\right)^2 , \tag{20}$$

where i indexes the number of neurons in the output layer. An iterative procedure for training, known as the backpropagation method, is given by Lapedes and Farber[25] as follows:

$$\Delta T_{ij} = \sum_p \varepsilon \delta_i^{(p)} O_j^{(p)} \tag{21}$$

and

$$\Delta\theta_i = \varepsilon \sum_p \delta_i^{(p)}, \tag{22}$$

where, for a neuron in the output layer,

$$\delta_i^{(p)} = \left(t_i^{(p)} - O_i^{(p)}\right) O_j^{(p)} \left(1 - O_i^{(p)}\right) \tag{23}$$

and, for a neuron in a hidden layer,

$$\delta_i^{(p)} = O_j^{(p)} \left(1 - O_i^{(p)}\right) \sum_j T_{ij}\, \delta_j^{(p)}. \tag{24}$$

This procedure involves first computing δ_i for the output layer and then using the previous equation to compute δ_i for the hidden layers.

Lapedes and Farber[25] proposed that a large class of functions of the form R^n mapping to R^m can be accurately approximated using only two hidden layers of neurons. They indicated that most signal processing tests cannot distinguish between chaotic behavior (nonlinear systems) and stochastic noise, and showed the ability of a neural network to model the Glass-Mackey equation, which exhibits chaotic behavior. For this test they used $g(x) = 1/2(1 + \tanh(x))$ as the sigmoidal function. Using the results of Takens[26] their system used four input nodes, since the Glass-Mackey equation generates a strange attractor with dimension 3.5.

In an earlier paper, Lapedes and Farber[27] showed through testing that neural networks are able to predict points in a chaotic time series with orders of magnitude greater accuracy than conventional methods, such as the Linear Predictive method and the Gabor-Volterra-Weiner Polynomial method. They state that neural networks perform well, since they globally approximate a system's mapping by performing a generalized mode decomposition. They mention that the accuracy of the network can be improved by increasing the number of neurons in the hidden layers (the layers between the input and output nodes). For the Mackey-Glass equation modeling, they used a two-layer neural network with 10 neurons in each hidden layer. This system required about 30 to 60 minutes training time on a Cray X-MP computer. They also presented an interesting analogy between neural networks and Fourier analysis,[27] which is summarized in the following paragraph.

Consider the sum of two sigmoids: $a_1 g\,(b_1 x + c_1) + a_2 g\,(b_2 x + c_2)$; it is seen that b adjusts the slope of the sigmoids, c adjusts the shifts, and a adjusts the gain. If we let g be sinusoid, then a acts like a Fourier amplitude, b like frequency, and c like phase shift. The a's are the synaptic weights of the hidden to output layer, the b's are synaptic

weights of the input to the hidden layer, and the number of g functions is the sum of number of hidden units in the hidden layer. The number of adjustable frequencies (using sine vice sigmoid) is thus determined by the number of neurons in the hidden layer.

Lapedes and Farber[27] also presented a convenient scaling methodology, where the network is "trained" using an input/output sequence in the range of 0 to 1. Once the network has completed training, the network's weights can then be scaled to handle inputs of an arbitrary range.

Many successful neural network applications to image processing have been reported. Widrow and Winter[5] used a neural network to produce a pattern recognition classifier that is insensitive to translation, rotation, and scale changes. Wilson[28] used neural networks as a voter for pattern recognition. Wilson applied vector morphology to the texture analysis problem and used neural networks for the voting logic involved in determining the "fit" of image areas to the set of possible structuring elements. Haykin and Leung[29] successfully modeled radar sea clutter using a two-layer neural network. Glover[30] described a system built for assembly line automatic inspection. The system consists of a video-input optical/electronic Fourier feature extraction module and a PC/AT with plug-in neural net board (Hecht-Nielson AZ1000 ANZA neurocomputer) for feature signature classification. This system performs global shape and texture analysis at speeds up to 15 images per second.

Tenorio and Hughes[31] discussed a system that uses a Markov image model, where Markov fields and approximate maximum a priori probabilities are input to a neural network that is used to segment the image. The system is invariant to rotation, scaling, position, translation, and multiplicity of objects. They indicated that the system correctly segments images regardless of the number of objects or their size, providing the objects are within the knowledge base of the network. This system will thus misinterpret unknown objects.

Mesrobian and Skrzypek[32] discussed a multilevel neural network approach to the discrimination of natural textures. Their proposed system will consist of three functional layers. The first layer is the feature extraction network, consisting of parallel elements to extract edges, line segments, line terminators, and corners from the image. The second layer is the local boundary detection layer that locates the perimeter of regions with uniform texture properties. The third and highest level layer is the higher order discrimination network that attempts to segment the textured images with a higher level of complexity. This level is based on the premise that grouping mechanisms employed in the discrimination of simple textures can also be used to discriminate textures of greater structural complexity.

Manjunath et al.[33] performed a comparison of texture segmentation algorithms using a Markov random field model, implemented on Hopfield neural networks. Segmentation tests were performed using an image consisting of six Brodatz[34] textures. The following misclassification error results were obtained:

- Maximum likelihood estimate — 22.2%
- Neural network with maximum likelihood estimate as initial state — 16.3%
- Neural network with random initial state — 14.7%

- Neural network with simulated annealing — 6.7%
- Maximizing the posterior marginal distribution — 7.1%
- Neural network with stochastic learning — 8.7%
- Hierarchial network (coarse to fine) — 8.2%.

They indicated that although simulated annealing worked the best, hundreds of iterations were required. Simulated annealing is a method that allows the system to diverge in a controlled fashion to prevent the solution from becoming trapped in a local minimum. The maximizing posterior marginal distribution rule also performed well but required hundreds of iterations. Reference 33 contains more details on the individual approaches used.

Neural networks have two extremely attractive features: they are conceptually simple, and they utilize a highly parallel approach that is well suited for parallel processing. Many elaborate neural network software packages are already commercially available, and neural network hardware will soon be available that will allow the implementation of high-speed networks. Neural networks may provide two promising avenues to the texture analysis problem. One method will use traditional texture analysis methods to extract feature vectors from an image and then use a neural network to make inferences based on these vectors. Another method will use a neural network to directly model the image by using the weights of the network as the feature vector of the image.

6.0 Modeling and Stochastic Techniques

Many other techniques have been proposed and attempted with various degrees of complexity and fairly uniform degree of success. These techniques can be roughly divided into modeling techniques, where a model is specified for the generation or analysis of texture, and stochastic techniques, where image statistics are used for texture identification. This research focused on the newer techniques of fractals and neural networks, although a few modeling and stochastic techniques were covered in the process. The following two sections describe the papers that present these types of techniques.

6.1 Modeling

Bovik et al.[35] discussed texture analysis using local spatial filters, i.e., the Gabor function given by:

$$h(x, y) = g(x', y') \exp[2\pi j(Ux + Vy)], \tag{25}$$

where $(x',y') = (x \cos\phi + y \sin\phi, -x \sin\phi + y \cos\phi)$ and

$$g(x, y) = 1/2\pi\lambda\sigma^2 \exp\left[-\frac{(x/\lambda)^2 + y^2}{2\sigma^2}\right].$$

With their method, tunable Gabor filters are used to model an image, and the segmentation is performed using channel amplitude and phase comparisons. The Gabor filters have tunable orientation and radial frequency bandwidths, making them well suited for this purpose. They indicated that the channel amplitude response can be used to detect

boundaries between textures and that large variations in the channel phase response provide a way to detect discontinuities in texture phase. The resulting segmentation achieved a good resemblance to visual perception. The research performed in this paper was directed by physiological evidence that Gabor-shaped receptive fields are fundamental to the biological processing of texture.

Chellappa and Kashyap,[36] Chellappa and Shankar,[37,38] Chellappa et al.,[39] and Chellappa[40,41] discuss the use of a two-dimensional noncausal autoregressive model for the synthesis and classification of texture. Chellappa and Kashyap[36] began with a Gaussian Markov random field model given by

$$y(s) = \sum_{r \in N} \theta_r y(s + r) + \sqrt{\beta}\omega(s), \tag{26}$$

and to simplify image synthesis they modified the model to be

$$y(s) = \sum_{r \in N} \theta_r y(s \oplus r) + \sqrt{\beta}\omega(s),\ s \in \Omega, \tag{27}$$

where $\Omega = [(i, j),\ 0 \le i, j \le M - 1]$, θ_r and β are the model parameters, $\omega(s)$ is a Gaussian noise signal, and N is the neighbor set of pixels for the pixel $y(s)$. $\oplus$ is the sum modulo M operator. The modified model is said to have nearly the same second-order properties as the ideal model for a large image. The set of M^2 equations can be represented in a matrix vector form as

$$B(\theta)\mathbf{y} = \sqrt{\beta}\omega, \tag{28}$$

where $B(\theta)$ is a block circulant matrix, and $\mathbf{y}$ and ω are M^2 vectors derived from the arrays $y(s)$ and $\omega(s)$. Let the eigenvalues of $B(\theta)$ be given by μ. If

$$\mu_s = (1 - 2\theta^T \psi_s),\ s \in \Omega, \tag{29}$$

where $\psi_s = \text{col.}\left[\exp\left(-\sqrt{-1}\,\frac{2\pi}{M} s^T r\right), r \in N\right]$, then an image vector $\mathbf{y}$ can be synthesized by the following equations:

$$y = \sum_{s \in \Omega} (f_s x_s / \mu_s) + \alpha \mathbf{l} \tag{30}$$

$$\alpha = E(y(s)),\ x_s = \sqrt{\beta}\,\mathbf{f}_s^{*T} \frac{\omega}{M^2},\ \mathbf{l} = \text{col. } (1, 1, \ldots, 1) \tag{31}$$

$$\mathbf{f}_s = \text{col. } [\mathbf{t}_j,\ \lambda_i \mathbf{t}_j,\ \ldots,\ \lambda_i^{M-1}\mathbf{t}_j],\ M^2 - \text{vector} \tag{32}$$

$$\mathbf{t}_j = \text{col. } [1,\ \lambda_j,\ \ldots,\ \lambda_j^{M-1}],\ M - \text{vector} \tag{33}$$

$$\lambda_i = \exp\left[\sqrt{-12}\,\frac{\pi_i}{M}\right],\ s = (i,\ j)\,. \tag{34}$$

The algorithm requires $O(M^2 \log M)$ operations. An extremely attractive feature of this algorithm is the ability to control a specific set of parameters θ_r and β, which determine the appearance of the texture. In earlier work Chellappa and Kashyap tabulated the parameter values required to duplicate several Brodatz[34] textures, using as few as 16 parameters for a good representation. They also presented maximum likelihood methods for determining the parameters required to fit their model to a given texture.

Khotanzad[42] used methods similar to those of Chellappa. He used a simultaneous autoregressive model given by:

for $\{g\ (x,\ y);\ x,\ y = 0, \ldots,\ M-1\}$

$$g(x,\ y) = \sum_{(i,j)\in N} \theta_{i,j}\ g(x \oplus i,\ y \oplus j) + \sqrt{\rho_N}\,\omega\,(x,\ y)\,, \tag{35}$$

where N is a neighbor set defined in the spatial domain. He stated that he used Chellappa's method of maximum likelihood for the model's parameter estimation. Khotanzad specifically addressed texture classification in his paper, using different N models. Using one N model, consisting of the four horizontal and vertical neighbors, and a second that uses the four diagonal neighbors, he obtained an average correct classification rate of 98%.

6.2 Stochastic

Cano et al.[43] proposed a method to find a set of texture parameters that is visually complete and compact, using a hierarchical filter bank approach (multiresolution). The process involves applying a local mask H:

$$G_1^{\,i}(x,\ y) = H_1^{\,i}(x,\ y)\ G_0\ (x,\ y);\ i = 1, \ldots,\ N\,, \tag{36}$$

where the first H is a low pass filter and successive H's are generated by

$$H_l^i = H_{l-1}^i \otimes 1_n\,. \tag{37}$$

In equation 37, $\otimes$ denotes the Kronecker product. The size of the mask increases geometrically with the level of resolution. The mean,

variance, and third-degree moments are then computed for each of the images G_1. The number of features that must be computed is given by

$$N_f = \ell n^2 \ (D - 1) + D \ , \tag{38}$$

where L is the number of hierarchical levels, D is the maximum degree of moment, and n is the size of the mask. This method worked well for stochastic textures but poorly for highly structured textures, since phase information is excluded in the filter bank technique. For highly structured textures Cano et al. developed a translation invariant operator based on the Fourier transform. The resulting operator worked well on periodic textures.

Fan[44] used an edge-based hierarchical algorithm for image segmentation. Generalized likelihood ratio like functions were used as discriminant functions, and boundaries were located with a maximum likelihood estimator. This algorithm required no prior knowledge of the texture model parameters or the number of texture regions. The method worked as well as 90% for some textures but as poorly as 60% on others.

Modestino et al.[1] discussed a texture discrimination approach based on spatial gray-level co-occurrences and maximum likelihood classification using a log-likelihood discriminator. This approach proved to be effective on random fields with identical second moments, where autocorrelation-power spectral density and edge density-correlation techniques were ineffective. Because autoregressive models cannot account for edges and because stochastic models do not provide for repetition of a local pattern, Modestino et al. used a discriminator that employs both correlation and edge density information. However, the discrimination process required a knowledge of the model parameters for each texture. Another disadvantage in this method is that an "interference" variable must be judiciously set by the user. A tradeoff exists for the strength of the interference between proper classification and ill-defined points of the intersection of region boundaries.

Vickers and Modestino[45] extended Modestino's[1] work. Using this technique, a training set of images is used for each texture class before the unknown set is applied to the classifier. Required preprocessing of the image included normalization, enhancement, and noise cleaning. Testing on 8-bit Brodatz[34] images yielded classification rates as high as 98% over a small data set. Vickers and Modestino[46] previously outlined a method to estimate the model parameters, whereas Modestino's[1] paper required a priori knowledge of these parameters.

7.0 Conclusions

It appears from this review that no single approach provides a robust texture analysis methodology without an overwhelming amount of complexity. The best approach to the problem seems to be to use a variety of these methods in their simplest and most computationally economic form. The idea of a multifaceted approach to texture analysis is also

supported by the design of the primate visual system. It has been found[47] that the visual center of the primate brain contains multiple maps of the visual field, each sensitive to particular aspects, such as motion, color, and shape.

Consequently, a texture analysis toolkit needs to be formed, preferably containing the better understood approaches. Edge detection must be included in the toolbox for texture analysis, since most of the texture approaches have difficulty with edges in an image. The edges typically need to be isolated so that texture analysis techniques can be applied to the regions between edges. The basic toolbox should probably include edge detection, the gray-level co-occurrence matrix and its derivative properties, fractal dimension, and a Markov random field model, since results obtained by using these approaches are well documented. With these analysis tools several image features can be extracted. These extracted features can then be used as input to a human operator, to an expert system, or to a neural network to perform the task of image interpretation.

Neural networks may ultimately provide the infrastructure for a "complete" visual system, incorporating both feature extraction and image interpretation. Progress continues to be made both in the understanding of the human visual system and in the development of parallel computing machines. With the successful alliance of these two fields of research, engineers and scientists may eventually be able to mimic the functionality of the human visual system.

8.0 References

1. Modestino, J. W., R. W. Fries, and A. L. Vickers (1980). Texture Discrimination Based Upon an Assumed Stochastic Texture Model. Rensselaer Polytechnic Inst., Dept. of Electrical and Systems Engineering, Technical Report, March.

2. Mandelbrot, B. (1983). *The Fractal Geometry of Nature.* New York (NY): W. H. Freeman and Company.

3. Ait-Kheddache, A. and S. A. Rajala (1988). Texture classification based on higher order fractals. *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, pp. 1112–1115, IEEE, April.

4. Hentschel, H. and I. Procaccia (1983). The infinite number of generalized dimensions of fractals and strange attractors. *Physica* 8D:435–444.

5. Widrow, B. and R. Winter (1979). Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*:25–39, March.

6. Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE* 67, May.

7. Conners, R. W. and C. A. Harlow (1980). A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-2, May.

8. Haralick, R. M., K. Shanmugam, and I. Dinstein (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics* 3:610–621, November.

9. Weszka, J., C. Dyer, and A. Rosenfeld (1976). A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics* 6:269–285, April.

10. Pentland, A. P. (1984). Fractal-based description of natural scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6:661–674, November.

11. Barnsley, M. F., R. L. Devaney, B. B. Mandelbrot, H. O. Peitgen, D. Saupe, and R. F. Voss (1988). *The Science of Fractal Images*. Berlin (Germany): Springer-Verlag.

12. Barnsley, M. (1988). *Fractals Everywhere*. Boston (MA): Academic Press.

13. Peitgen, H. and P. Richter (1986). *The Beauty of Fractals*. Berlin (Germany): Springer-Verlag.

14. Brammer, R. (1989). Unified image computing based on fractals and chaos model techniques. *Optical Engineering* 28:726–734, July.

15. Fournier, A., D. Fussell, and L. Carpenter (1982). Computer rendering of stochastic models. *Communications of the ACM* 25:371–384, June.

16. Keller, J. M., S. Chen, and R. M. Crownover (1989). Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and Image Processing* 45:150–166, February.

17. Margerum, E. and A. Werkheiser (1988). Computer Generation of Fractal Terrains. Army Engineer Topographic Lab., Fort Belvoir, VA, Technical Report, September.

18. Vernazza, G. (1987). Performances of different fractal-dimension estimators of natural image texture. *Proceedings of the IASTED International Symposium*, pp. 138–142, June.

19. Peleg, S., J. Naor, R. Hartley, and D. Avnir (1984). Multiple resolution texture analysis and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6:518–523, July.

20. Arduini, F., C. Dambra, S. Dellepiane, S. B. Serpico, G. Vernazza, and R. Viviani (1988). Fractal dimension estimation by adaptive mask selection. *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, pp. 1116–1119, IEEE, April.

21. Ait-Kheddache, A. and S. A. Rajala (1988). Texture classification based on higher order fractals. *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, pp. 1112–1115, IEEE, April.

22. Szeliski, R. and D. Terzopoulos (1989). From splines to fractals. *ACM Computer Graphics* 23:51–60, July.

23. Ahlberg, J., E. Nilson, and J. Walsh (1967). *The Theory of Splines and their Applications*. New York (NY): Academic Press.

24. Bartels, R. H., J. C. Beatty, and B. A. Barsky (1987). *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Los Altos (California): Morgan Kaufmann.

25. Lapedes, A. and R. Farber (1988). *How Neural Networks Work*. Neural Information Processing Systems, New York (NY): American Institute of Physics Press, pp. 442–456.

26. Takens, F. (1981). Detecting Strange Attractor in Turbulence. In *Lecture Notes in Mathematics*, D. Rand and L. Young (eds.), Berlin (Germany): Springer-Verlag, p. 366.

27. Lapedes, A. and R. Farber (1987). Nonlinear Signal Processing Using Neural Networks: Prediction and System Modelling. Los Alamos National Lab, Los Alamos, NM, June, Technical Report, LA-UR-87-2662.

28. Wilson, S. S. (1989). Vector morphology and iconic neural networks. *IEEE Transactions on Systems, Man and Cybernetics* 19:1636–1644, November–December.

29. Haykin, S. and H. Leung (1989). Chaotic model of sea clutter using a neural network. In *Advanced Algorithms and Architectures for Signal Processing IV* 1152:18–21, SPIE.

30. Glover, E. (1988). Optical Fourier/electronic neurocomputer automated inspection system. In *Proceedings of the International Conference on Neural Networks*, San Diego, CA, pp. 569–576, IEEE, July.

31. Tenorio, M. F. and C. S. Hughes (1984). Real time noisy image segmentation using an artificial neural network model. *Proceedings of the 1st International Conference on Neural Networks* IV:357–363, San Diego, CA, IEEE, June.

32. Mesrobian, E. and J. Skrzypek (1984). Discrimination of natural textures: a neural network architecture. *Proceedings of the 1st International Conference on Neural Networks* IV:247–255, San Diego, CA, IEEE, June.

33. Manjunath, B., T. Simchony, and R. Chellappa (1990). Stochastic and deterministic networks for texture segmentation. *IEEE Transactions on Acoustics, Speech and Signal Processing* 38:1039–1049, June.

34. Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. New York (NY): Dover.

35. Bovik, A., M. Clark, and W. Geilser (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12(1):55–73.

36. Chellappa, R. and R. L. Kashyap (1985). Texture synthesis using 2-D noncausal autoregressive models. *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-33:194–203, February.

37. Chellappa, R. and C. Shankar (1984). Statistical approaches for texture synthesis. *Proceedings of the Fifth Annual Conference and Exposition—Computer Graphics '84* 2:671–678, National Computer Graphics Association, May.

38. Chellappa, R. and C. Shankar (1985). Classification of textures using Gaussian Markov random fields. *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-33(4):959–963.

39. Chellappa, R., S. Chatterjee, and R. Bagdazian (1985). Texture synthesis and compression using Gaussian Markov random field models. *IEEE Transactions on Systems, Man and Cybernetics* SMC-15:298–303, March–April.

40. Chellappa, R. (1981). Fitting Markov Random Field Models to Images. Air Force Office of Scientific Research, Technical Report, January.

41. Chellappa, R. (1981). Synthesis of Textures Using Simultaneous Autoregressive Models. Air Force Office of Scientific Research, Technical Report, AFOSR-TR-81-0795, July.

42. Khotanzad, A. (1987). Recognition of textured images using model based features selected via synthesis. *Proceedings of the IASTED International Symposium*, pp. 18–22, May.

43. Cano, D., T. Ha Minh, and M. Kunt (1988). Texture analysis and synthesis. In *From Pixels to Features: Proceedings of a Workshop*, pp. 127–140, August.

44. Fan, Z. (1988). Segmentation and Classification of Textured Images. Ph.D dissertation, University of Rhode Island, Kingston, RI.

45. Vickers, A. and J. Modestino (1982). Maximum likelihood approach to texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-4:61–68, January.

46. Vickers, A. L. and J. W. Modestino (1981). Further Results on Texture Discrimination Based Upon an Assumed Stochastic Texture Model. Rensselaer Polytechnic Inst., Dept. of Electrical and Systems Engineering, Technical Report, March.

47. Montgomery, G. (1991). The mind's eye. *Discover*, pp. 51–56, May.

Distribution List

Applied Physics Laboratory
Johns Hopkins University
Johns Hopkins Road
Laurel MD 20707

Applied Physics Laboratory
University of Washington
1013 NE 40th St.
Seattle WA 98105

Applied Research Laboratory
Pennsylvania State University
P.O. Box 30
State College PA 16801-0030

Applied Research Laboratory
University of Texas at Austin
P.O. Box 8029
Austin TX 78713-8029

Assistant Secretary of the Navy
Research, Development & Acquisition
Navy Department
Washington DC 20350-1000

Chief of Naval Operations
Department of the Navy
Washington DC 20350-2000
Attn: OP-71
OP-987

Chief of Naval Operations
Oceanographer of the Navy
U.S. Naval Observatory
34th & Massachusetts Ave. NW
Washington DC 20392-1800
Attn: OP-096
OP-096B

Defense Mapping Agency
8613 Lee Hwy.
Fairfax VA 22031-2138
Attn: Code PRN, Mailstop A-13
Director

Fleet Antisub Warfare Tng Ctr-Atl
Naval Station
Norfolk VA 23511-6495
Attn: Commanding Officer

Fleet Numerical Oceanography Center
Monterey CA 93943-5005
Attn: Commanding Officer

National Ocean Data Center
1825 Connecticut Ave., NW
Universal Bldg. South, Rm. 206
Washington DC 20235

Naval Air Systems Command HQ
Washington DC 20361-0001
Attn: Commander

Naval Air Warfare Center
Aircraft Division Warminster
Warminster PA 18974-5000
Attn: Commander

Naval Civil Engineering Laboratory
Port Hueneme CA 93043
Attn: Commanding Officer

Naval Command Control and Ocean
Surveillance Center
RDT&E Division
San Diego CA 92152-5000
Attn: Commander

Naval Facilities Engineering Command
200 Stovall St.
Alexandria VA 22332-2300
Attn: Commander

Naval Oceanographic Office
Stennis Space Center MS 39522-5001
Attn: Commanding Officer
Code TD
Library (2)

Naval Oceanography Command
Stennis Space Center MS 39529-5000
Attn: Commander

Naval Postgraduate School
Monterey CA 939-3
Attn: Superintendent

Naval Research Laboratory
Atmospheric Directorate
Monterey CA 93943-5006
Attn: Director
Code 400

Naval Research Laboratory
Stennis Space Center MS 39529-5004
Attn: Code 115
Code 125L (10)
Code 125P
Code 200
Code 300

Naval Research Laboratory
Washington DC 20375
Attn: Commanding Officer
Library (2)

Naval Sea Systems Command HQ
Washington DC 20362-5101
Attn: Commander

Naval Surface Warfare Center
Dahlgren Division
Detachment White Oak
10901 New Hampshire Ave.
Silver Spring MD 20903-5000
Attn: Officer in Charge
Library

Naval Surface Warfare Center
Dahlgren Division
Dahlgren VA 22448-5000
Attn: Commander

Naval Surface Warfare Center
Coastal Systems Station
Dahlgren Division
Panama City FL 32407-5000
Attn: Commanding Officer

Naval Surface Warfare Center
Carderock Division
Bethesda MD 20084-5000
Attn: Commander

Naval Undersea Warfare Center
Division
Newport RI 02841-5047
Attn: Commander

Naval Undersea Warfare Center Det
New London CT 06320
Attn: Officer in Charge

Office of Naval Research
800 N. Quincy St.
Arlington VA 22217-5000
Attn: Code 10D/10P, E. Silva
Code 112, E. Hartwig
Code 12
Code 10

Office of Naval Research
ONR European Office
PSC 802 Box 39
FPO AE 09499-0700
Attn: Commanding Officer

Office of Naval Technology
800 N. Quincy St.
Arlington VA 22217-5000
Attn: Code 20, P. Selwyn
Code 228, M. Briscoe
Code 228, CDR L. Bounds
Code 22, T. Warfield

Scripps Institution of Oceanography
University of California
291 Rosecrans St.
San Diego CA 92106-3505

Scripps Institution of Oceanography
P.O. Box 6049
San Diego CA 92166-6049

Space & Naval Warfare Sys Com
Director of Navy Laboratories
SPAWAR 005
Washington DC 20363-5100
Attn: Commander

Woods Hole Oceanographic Institution
P.O. Box 32
Woods Hole MA 02543
Attn: Director

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*	2. Report Date.	3. Report Type and Dates Covered.
	January 1992	Final

4. Title and Subtitle.

Survey of Texture Segmentation, Classification, and Synthesis Methods

5. Funding Numbers.

Job Order No.	13512A
Program Element No.	0602435N
Project No.	RM35G85
Task No.	801
Accession No.	DN255031

6. Author(s).

Brian S. Bourgeois, Dr. Charles L. Walker

7. Performing Organization Name(s) and Address(es).

Naval Oceanographic and Atmospheric Research Laboratory
Ocean Science Directorate
Stennis Space Center, Mississippi 39529-5004

8. Performing Organization Report Number.

NOARL Report 30

9. Sponsoring/Monitoring Agency Name(s) and Address(es).

Naval Oceanographic and Atmospheric Research Laboratory
Ocean Science Directorate
Stennis Space Center, Mississippi 39529-5004

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution/Availability Statement.

Approved for public release; distribution is unlimited. Naval Oceanographic and Atmospheric Research Laboratory, Stennis Space Center, Mississippi 39529-5004

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

This report reviews the literature in the areas of image texture segmentation, classification, and synthesis methods. The approaches to these areas are grouped into areas of fractal, spline, neural networks, modeling, and stochastic methods. An immense amount of literature was reviewed, and techniques with the most merit are presented. From the review it appears that no single approach provides a robust texture analysis methodology without requiring overwhelming complexity. It seems that the best approach to the problem may be to use a variety of these methods in the simplest and most computationally economic forms.

14. Subject Terms.

hydrography, bathymetry, optical properties, tides, reverberation

15. Number of Pages.

24

16. Price Code.

17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
Unclassified	Unclassified	Unclassified	None